

Algoritma *Guided Greedy Randomized Adaptive Search Procedures with Fixed Threshold* Untuk Keseimbangan Lintasan Perakitan *Mixed Model* dengan Kriteria Minimisasi Jumlah Stasiun Kerja*

FAJAR ULMA, ALEX SALEH, EMSOSFI ZAINI.

Jurusan Teknik Industri

Institut Teknologi Nasional (ITENAS) Bandung

Email: fajarulma@gmail.com

ABSTRAK

Assembly line balancing (ALB) adalah suatu penugasan pekerjaan pada stasiun kerja yang saling berhubungan antar lintasan produksi untuk memperoleh target laju produksi yang terpenuhi. Pembahasan yang dilakukan pada penelitian ini adalah tentang keseimbangan lintasan perakitan menggunakan lintasan mixed model. Penelitian ini bertujuan untuk mengembangkan algoritma Guided GRASP With Fixed Threshold terhadap lintasan perakitan mixed model dengan kriteria minimisasi jumlah stasiun kerja. Pengujian algoritma usulan dilakukan dengan menggunakan tiga skenario, skenario 1 menggunakan set data Bedworth dan Bailey (1982) yang terdiri dari 2 variasi produk dengan jumlah elemen kerja (n) = 11, skenario 2 menggunakan set data Thomopoulos (1970) yang terdiri dari 3 variasi produk dengan jumlah elemen kerja (n) = 19, dan skenario 3 menggunakan set data penelitian Su dan Lu (2007) yang terdiri dari 3 variasi produk dengan jumlah elemen kerja (n) = 17. Hasil pengujian ketiga skenario menunjukkan bahwa algoritma usulan dapat memberikan solusi yang lebih baik pada efisiensi lintasan dan smoothness index dengan jumlah stasiun kerja yang sama bila dibandingkan dari penelitian-penelitian sebelumnya yang telah dipublikasikan.

Kata Kunci: Keseimbangan Lintasan, Mixed Model, Guided GRASP With Fixed Threshold

ABSTRACT

Assembly line balancing (ALB) is a job placement assignment on workstations that are interconnected across the track to get the target production rate of production is met. The discussion carried out in this study is about the balance of mixed model track assembly. The study aims to develop Guided GRASP With Fixed threshold algorithm to address assembly line mixed model with a minimization criterion number of work stations. The algorithm is then tested using three three scenarios, scenario 1 using data sets Bedworth and Bailey (1982) which consists of 2 variations of the product by the number of working elements (n) = 11, scenario 2 using data sets Thomopoulos (1970) which consists of 3

* Makalah ini merupakan ringkasan dari Tugas Akhir yang disusun oleh penulis pertama dengan pembimbingan penulis kedua dan ketiga. Makalah ini merupakan draft awal dan akan disempurnakan oleh para penulis untuk disajikan pada seminar nasional dan/atau jurnal nasional

variations product by the number of working elements (n) = 19, and scenario 3 using data sets S_u and L_u (2007) which consists of 3 variations of the product by the number of working elements (n) = 17. The results obtained show that the proposed algorithm can provide better solution to efficiency line and smoothness index with the same number of work stations when compared to previous studies that have been published.

Keywords: *Line Balancing, Mixed Model, Guided GRASP With Fixed Threshold.*

1. PENDAHULUAN

Assembly line balancing (ALB) adalah suatu metode penugasan sejumlah pekerjaan ke dalam stasiun-stasiun kerja yang saling berkaitan dalam satu lintasan produksi sehingga setiap stasiun kerja memiliki waktu yang tidak melebihi waktu siklus dari stasiun kerja tersebut. Satu set elemen kerja ($j=1,2,3,\dots,n$) ditempatkan pada satu set stasiun kerja untuk dirakit dengan memperhatikan hubungan preseden yang diwakili oleh *precedence diagram*. *Precedence diagram* adalah diagram yang menggambarkan hubungan atau saling keterkaitan antara satu pekerjaan dengan pekerjaan lainnya.

Simple assembly line balancing problem (SALBP) merupakan lintasan perakitan tunggal klasik. Menurut Scholl dan Becker (2006) secara umum permasalahan keseimbangan lintasan perakitan dapat dikelompokkan menjadi permasalahan keseimbangan lintasan perakitan Tipe I (*simple assembly line balancing problem Type I, SALBP I*), permasalahan keseimbangan lintasan perakitan Tipe II (*simple assembly line balancing problem Type II, SALBP II*), SALBP-E, dan SALBP-F.

SALBP merupakan suatu lintasan perakitan tunggal klasik yang salah satu ciri-cirinya adalah bersifat produksi massal untuk menghasilkan suatu produk homogen yang memiliki n operasi yang telah diketahui. Dari salah satu ciri-cirinya tersebut, SALBP memiliki kekurangan dalam memenuhi keinginan konsumen yang sangat beragam. Fenomena tersebut dapat diamati dalam peningkatan jumlah model mobil, TV, komputer, dan lain sebagainya. Sebagian besar model memiliki sejumlah fitur, dan pelanggan dapat memilih model yang didasarkan pada keinginannya. Berarti bahwa fitur yang berbeda merupakan bagian-bagian tambahan yang harus ditambahkan pada model dasar sesuai keinginan konsumen. Biaya yang tinggi untuk membangun dan memelihara jalur perakitan, produsen memproduksi satu model dengan fitur berbeda atau beberapa model pada jalur perakitan tunggal. Dalam keadaan ini, *mixed model assembly line balancing* (MALB) muncul untuk kelancaran produksi dan penurunan biaya akibat tuntutan untuk model yang berbeda dan untuk fitur bervariasi setiap hari. MALB memperlihatkan lintasan perakitan modern yang lebih realistis, dimana permintaan ditandai dengan variabilitas yang tinggi dan *volume* yang relatif kecil untuk setiap modelnya.

Metode yang digunakan dalam penelitian ini adalah algoritma *Guided Greedy Randomized Adaptive Search Procedures (Guided GRASP) With Fixed Threshold*. *Guided GRASP With Fixed Threshold* merupakan pengembangan algoritma dari metode *Guided GRASP*. *Guided GRASP* merupakan pengembangan dari metode GRASP dimana GRASP pertama kali diusulkan oleh Feo dan Resende (1995) untuk memecahkan masalah penjadwalan pengemudi bus dan *travelling salesman problem*. Algoritma *Guided GRASP With Fixed Threshold* merupakan pengembangan algoritma dari metode *Guided GRASP* dengan penambahan nilai *threshold*. *Threshold* pertama kali dikembangkan oleh Dueck dan Scheuer (1990) untuk menyelesaikan masalah komputasi. Penambahan fungsi *threshold* bertujuan agar mendapatkan solusi yang

lebih baik karena solusi-solusi buruk yang berada dalam batas *threshold* masih dapat dipertimbangkan untuk menjadi solusi yang lebih baik selanjutnya, hal ini berbeda dengan metode sebelumnya yang hanya mempertimbangkan solusi-solusi yang lebih baik. Penelitian MALBP menggunakan algoritma *Guided GRASP With Fixed Threshold* belum pernah dilakukan sebelumnya, sehingga perlu dilakukan pengembangan algoritma untuk pemecahan MALBP menggunakan algoritma *Guided GRASP With Fixed Threshold* dengan kriteria minimisasi jumlah stasiun kerja dengan performansi maksimisasi efisiensi lintasan.

Pada penelitian ini, sistem yang akan dibahas adalah model lintasan perakitan *mixed model. Mixed model assembly line balancing* (MALB) merupakan beberapa variasi dari suatu produk dasar yang dirakit pada satu lintasan yang sama. Perbedaan dari setiap variasi ini dapat didasarkan pada ukuran, warna, penggunaan material atau peralatan. Metode dasar pemecahan masalah yang digunakan pada penelitian ini adalah algoritma *guided greedy randomized adaptive search procedures (Guided GRASP) with fixed threshold* dengan kriteria minimisasi jumlah stasiun kerja dengan performansi maksimisasi efisiensi lintasan. Penelitian ini bertujuan untuk mengembangkan model algoritma *guided greedy randomized adaptive search procedures (Guided GRASP) with fixed threshold* pada lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja dengan performansi maksimisasi efisiensi lintasan.

Batasan-batasan masalah yang dilakukan pada penelitian ini adalah:

1. Jumlah operator yang diperlukan untuk merakit produk tidak dibahas.
2. Peralatan dan mesin yang digunakan untuk merakit produk tidak dibahas.
Asumsi-asumsi yang digunakan dalam penelitian ini adalah:
 1. Waktu proses setiap elemen kerja telah diketahui secara deterministik.
 2. Tidak diperbolehkan terdapat *buffer* antara stasiun.
 3. Waktu *set-up* termasuk ke dalam waktu proses.

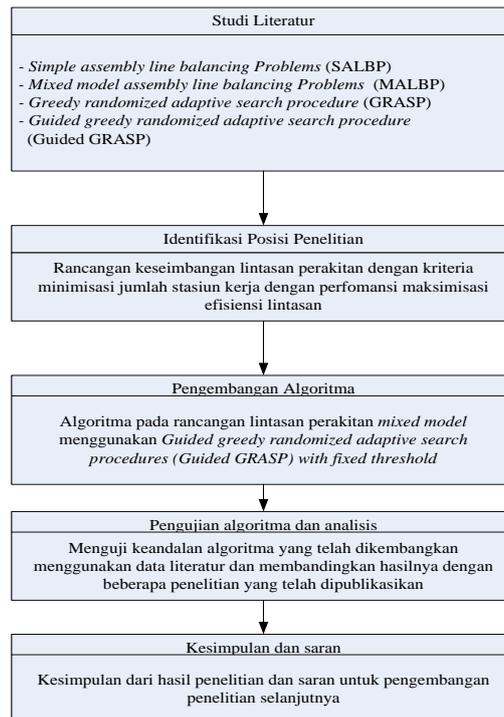
2. METODOLOGI PENELITIAN

Metodologi penelitian merupakan suatu penjelasan mengenai langkah-langkah yang harus dilakukan dalam proses pengembangan algoritma. Hasil yang didapatkan akan terstruktur dan terarah dalam proses pemecahan masalah yang sesuai dengan tujuan yang akan dicapai. Tahap-tahap dalam metodologi penelitian ini dapat dilihat pada Gambar 1.

Penelitian ini diawali dengan melakukan studi literatur terhadap penelitian *simple assembly line balancing problem* (SALBP), *mixed model assembly line balancing* (MALB) yang diselesaikan dengan algoritma *Guided GRASP with fixed threshold*. Penelitian yang menjadi dasar dari penelitian ini adalah:

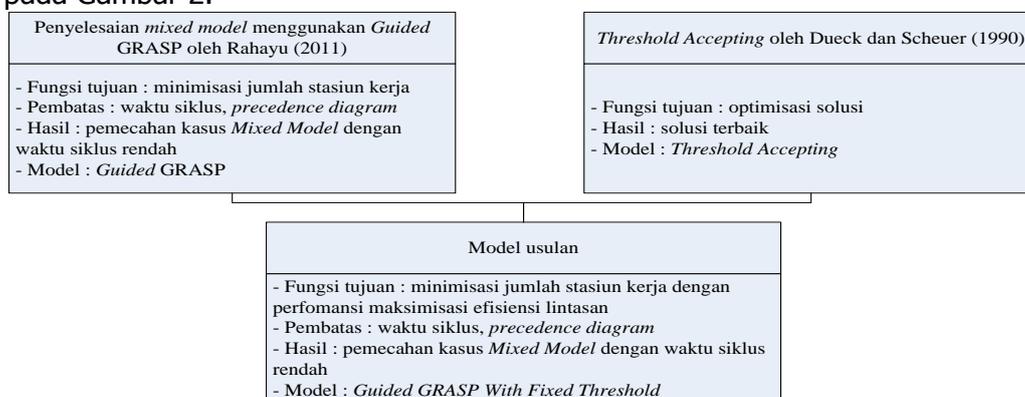
1. Thomopoulos (1970) melakukan penelitian pertama kali tentang permasalahan *mixed model assembly line balancing problems* (MALBP) menggunakan metode *heuristic* dengan kriteria minimisasi jumlah stasiun kerja.
2. Gocken dan Erel (1997) melakukan penelitian tentang permasalahan *mixed model assembly line balancing problems* (MALBP) menggunakan optimisasi dengan kriteria minimisasi jumlah stasiun kerja.
3. Permasalahan ALB menjadi 4 bagian yaitu SALBP I, SALBP II, SALBP-E, dan SALBP-F oleh Scholl dan Becker (2006).
4. Haq et al. (2006) melakukan penelitian tentang permasalahan *mixed model assembly line balancing problems* (MALBP) menggunakan optimisasi dengan kriteria minimisasi jumlah stasiun kerja.

5. Putrianti (2011) melakukan penelitian tentang permasalahan *mixed model assembly line balancing problems* (MALBP) menggunakan metode *Greedy Randomized Adaptive Search Procedures* (GRASP) dengan kriteria minimisasi jumlah stasiun kerja.
6. Rahayu (2011) melakukan penelitian tentang permasalahan *mixed model assembly line balancing problems* (MALBP) menggunakan metode *Guided Greedy Randomized Adaptive Search Procedures* (Guided GRASP) dengan kriteria minimisasi jumlah stasiun kerja.
7. Pengembangan algoritma *Guided GRASP* dengan menambahkan fungsi *fixed threshold*. *Threshold accepting* pertama kali dikembangkan oleh Dueck dan Scheuer (1990).



Gambar 1. Metodologi Penelitian

Penelitian ini merupakan pengembangan model dari penelitian yang telah dilakukan sebelumnya, yaitu Rahayu (2011) yang mengembangkan model *Guided GRASP* pada lintasan *mixed model* dengan tujuan minimasi jumlah stasiun kerja. Metode yang dikembangkan oleh Rahayu (2011) dikombinasikan dengan penambahan fungsi *threshold*. Fungsi *threshold* bertujuan untuk memperoleh solusi yang lebih baik. Skema pengembangan algoritma dapat dilihat pada Gambar 2.



Gambar 2. Skema Pengembangan Algoritma

Pengujian algoritma ini dilakukan untuk mengetahui performansi dari algoritma usulan dengan menggunakan 3 set data.

3. PENGEMBANGAN ALGORITMA

Pada sub bab ini akan dibahas mengenai kerangka pengembangan algoritma untuk memecahkan permasalahan *mixed model* melalui tahapan algoritma dasar dan algoritma usulan.

3.1 Algoritma Dasar

Model dasar untuk penelitian ini merupakan pengembangan dari beberapa penelitian diantaranya adalah Rahayu (2011) menyelesaikan MALBP dengan metode *guided GRASP* dan metode *threshold accepting* yang pertama kali dikembangkan oleh Dueck dan Scheuer (1990). Tujuan pengembangan model ini adalah untuk mendapatkan rancangan lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja dengan performansi maksimisasi efisiensi lintasan perakitan menggunakan algoritma usulan yang diadaptasi dari *guided greedy randomized adaptive search procedure* (GRASP).

3.2 Algoritma Usulan

Guided greedy randomized adaptive search procedures (*Guided GRASP*) *With Fixed Threshold* terbagi menjadi dua tahapan yaitu konstruksi solusi inisial dan pencarian lokal. Notasi-notasi yang digunakan pada algoritma usulan adalah sebagai berikut:

Parameter:

- i = indeks untuk stasiun kerja (SK); ($i = 1, 2, \dots, m$)
- z = indeks untuk variasi produk; ($z = 1, 2, \dots, z$)
- t_j = waktu proses pada elemen kerja j
- d_j = demand untuk setiap variasi produk
- CT = waktu siklus
- k = indeks iterasi pada local search; ($k = 1, 2, \dots, kmax$)
- p = indeks iterasi keseluruhan ($p = 1, 2, \dots, pmax$)
- α = threshold parameter; $\alpha \in U[0.5, 1]$

Variabel:

- V = himpunan elemen kerja yang dapat ditambahkan (assignable)
- V_c = himpunan elemen kerja yang dijadikan kandidat prospektif pemilihan RCL
- V_a = himpunan elemen kerja yang telah ditambahkan ke dalam SK (assigned)
- ST_i = akumulasi waktu elemen kerja pada SK_i
- ST_{iz} = akumulasi waktu elemen kerja pada SK_i pada setiap variasi produk
- Δ = selisih waktu siklus dengan akumulasi waktu elemen kerja pada SK_i
- EL_k = efisiensi lintasan pada iterasi k
- SI_k = *smoothness index* pada iterasi k
- EL_p = efisiensi lintasan pada iterasi p
- SI_p = *smoothness index* pada iterasi p
- $h(j)$ = nilai fungsi *greedy* setelah penambahan elemen $j \in RCL$
- $\underline{h}(j)$ = total waktu SK minimum setelah penambahan elemen kerja ke- j
- $\overline{h}(j)$ = total waktu SK maksimum setelah penambahan elemen kerja ke- j

Algoritma usulan *Guided GRASP with fixed threshold* untuk menyelesaikan masalah lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja dengan performansi maksimisasi efisiensi lintasan adalah sebagai berikut:

Tahap 1 – Initial Solution

Langkah 0

Input data: $i, j, t_j, d, CT, V, V_c = \emptyset$, precedence diagram masing-masing variasi.

Langkah 1

Lakukan proses penggabungan *precedence diagram* dengan menggunakan *precedence matrices*.

Langkah 2

set $p_{max} = \max(2, \lceil n/10 \rceil$)

Langkah 3

Set $p = 1$ dan Set $i = 1$

Langkah 4

a. Perbaharui kandidat prospektif (V_c)

Pilih elemen kerja $j \in V$ yang dapat dijadikan kandidat untuk ditambahkan ke dalam SK ke- i dengan memperhatikan *precedence relations*.

b. Periksa apakah $V_c = 1$?

Jika ya, lanjutkan ke Langkah 6; dan tempatkan elemen kerja j ke dalam V_a pada SK ke- i .

Lainnya, lanjutkan ke Langkah 5.

Langkah 5

a. Tentukan ST_i minimum dengan Persamaan (3.1) dan maksimum dengan Persamaan (3.2)

b. Bangkitkan secara *random* $\alpha \in U[(0,5), 1]$

c. Bentuk RCL dengan menggunakan Persamaan (3.3)

d. Set L_{min} dengan menggunakan Persamaan (3.4)

e. Periksa apakah $RCL < L_{min}$?

Jika ya, lakukan proses *relaxing*; urutkan $j \in V_c$ dari t_j terkecil ke terbesar, pilih urutan sebanyak L_{min} untuk masuk ke dalam RCL dan lanjutkan ke Langkah 6. Lainnya, lanjutkan ke Langkah 6.

Langkah 6

Hitung ST_{iz} setiap elemen kerja j pada RCL dan periksalah apakah $ST_{iz} > CT$ bila ditambahkan dengan elemen kerja tersebut?

Jika ya, perbaiki RCL dan lanjutkan ke Langkah 7;

Lainnya lanjutkan ke Langkah 7.

Langkah 7

Hitung $\Delta = CT - ST_i$ setiap elemen kerja j pada RCL kemudian pilih elemen kerja dengan nilai Δ terkecil dan lanjutkan ke Langkah 8.

Langkah 8

Periksa apakah $ST_i < CT$ dan atau $ST_m < CT$?

Jika ya, tempatkan elemen kerja j ke dalam V_a pada SK ke- i ; lanjutkan ke Langkah 9.

Lainnya, set $i = i + 1$; lalu tempatkan elemen kerja j ke dalam V_a pada SK ke- i lanjutkan ke Langkah 9.

Langkah 9

Periksa apakah $|V| = 0$?

Jika ya, berhenti; lanjutkan ke Tahap 4 – *local search*.

Lainnya, kembali ke Langkah 4.

Tahap 2 – Local Search

Langkah 10

Input data: jalur lintasan perakitan pada Tahap 1.

Hitung efisiensi lintasan (EL_k) dan *Smoothness index* (SI_k) awal.

Hitung pembatas *threshold*, pembatas yang dipakai sebesar 5%, 15%, dan 25%
set $k_{max} = \max(4, \lceil n/4 \rceil)$

Langkah 11

Set $k = 1$

Langkah 12

a. Lakukan proses *exchange*.

a.1. Ambil dua buah elemen kerja j dari SK ke- i yang berbeda secara *random*.

a.2. Tukarkan kedua elemen kerja j tersebut.

Periksa apakah penukaran elemen kerja j melanggar preseden dan ST_i melebihi CT ?

Jika ya, batalkan, kembali ke Langkah 12a.1.;

Lainnya, tempatkan elemen kerja j pada SK ke- i , lakukan proses *exchange*

a.3. Apakah elemen kerja j masih dapat ditukar ke elemen kerja lainnya?

Jika ya, kembali ke langkah 12a.1

Lainnya, lanjut kelangkah 12a.4

a.4. Hitung efisiensi lintasan (EL_k) dan *Smoothness index* (SI_k)

a.5. Apakah $EL_{th} \leq EL_k \leq 100\%$

Jika ya, lanjut ke langkah 12b

Lainnya, kembali ke langkah 12a.1

b. Lakukan proses *insert*.

b.1. Ambil salah satu elemen kerja j dari SK ke- i secara *random*.

b.2. Pindahkan elemen kerja j ke SK_i ($i \neq j$)

Periksa apakah pemindahan elemen kerja j melanggar preseden dan ST_i melebihi CT ?

Jika ya, batalkan; kembali ke Langkah 12b.1;

Lainnya, tempatkan elemen kerja j pada SK ke- i , lakukan proses *insert*

b.3. Apakah elemen kerja j masih dapat *diinsert* ke stasiun kerja lainnya?

Jika ya, kembali ke langkah 12b.1

Lainnya, lanjut kelangkah 12b.4

b.4. Apakah terdapat solusi *insert* yang menyebabkan SK bertambah?

Jika ya, hitung Efisiensi lintasan. Apakah efisiensi lintasan 100%?

Jika ya, lanjut kelangkah 19

Lainnya, lanjut kelangkah 15 dengan solusi yang telah berubah.

Jika tidak, lanjut kelangkah 12.b.5

b.5. Hitung efisiensi lintasan (EL_k) dan *Smoothness index* (SI_k)

b.6. Apakah $EL_{th} \leq EL_k \leq 100\%$

Jika ya, lanjut ke langkah 13

Lainnya, kembali ke langkah 12b.1

Langkah 13

Periksa apakah $EL_k = 100\%$?

Jika ya, lanjutkan ke Langkah 17

Jika tidak, lanjut ke langkah 14

Langkah 14

Periksa apakah EL_k lebih besar atau sama dengan EL_k solusi inisial? , jika ya perbarui jalur lintasan hasil perhitungan iterasi (k) dan perbarui pembatas *threshold*.

Langkah 15

Periksa apakah $k = k_{max}$?

Jika ya, lanjutkan ke Langkah 15.

Lainnya, simpan nilai EL_k lalu set $k = k + 1$ dan kembali ke Langkah 12.

Langkah 17

Tampilkan rancangan lintasan SK perakitan *mixed model* terpilih beserta nilai EL_k

lalu set $\max EL_{k+1} = EL_p$ dan lanjutkan ke Langkah 18.

Langkah 18

a) Periksa apakah $p = p_{max}$?

Jika ya, tampilkan nilai EL_p dan pilih nilai $\max EL_p = \max \{EL_p\}$

Langkah 19

Tampilkan lintasan sebagai *final local optimal* untuk MALB beserta parameternya.

3.4 Perbedaan Algoritma Usulan Dengan Algoritma *Guided Grasp*

Algoritma *Guided GRASP With Fixed Threshold* merupakan pengembangan algoritma dari metode *Guided GRASP*. Perbedaan algoritma usulan dengan algoritma *Guided GRASP* terletak pada proses *local search*. Pada penelitian sebelumnya proses *local search* yang didalamnya terdapat proses *exchange* dan *insert* dilakukan secara bertahap dimulai melakukan proses *exchange* dengan menukar elemen kerja tertentu dengan elemen kerja lain dengan syarat tidak melanggar *precedence diagram* secara terus-menerus sampai tidak ada lagi elemen kerja yang bisa ditukar. Setelah melakukan proses *exchange* lakukan proses *insert* dengan cara yang sama dengan *exchange*. Kemudian hitung efisiensi lintasan dari konfigurasi yang didapat dari proses *exchange* dan *insert*, ulang kembali proses tersebut sampai mencapai iterasi yang telah ditentukan. Untuk algoritma usulan, proses *exchange* dilakukan dengan menukar satu elemen kerja dengan elemen kerja yang lain dengan syarat tidak melanggar *precedence diagram*. Kemudian dari konfigurasi yang dihasilkan *exchange* hitung besarnya efisiensi lintasan dan bandingkan dengan nilai *threshold* apakah nilai efisiensi yang didapat masih berada dalam ambang batas *threshold*. Selanjutnya lakukan proses *insert* dengan cara yang sama seperti proses *exchange*. Solusi yang dihasilkan oleh proses *insert* tersebut dijadikan solusi untuk proses iterasi selanjutnya yang telah ditentukan.

3. PENGUJIAN ALGORITMA DAN ANALISIS

4.1 Pengujian Algoritma

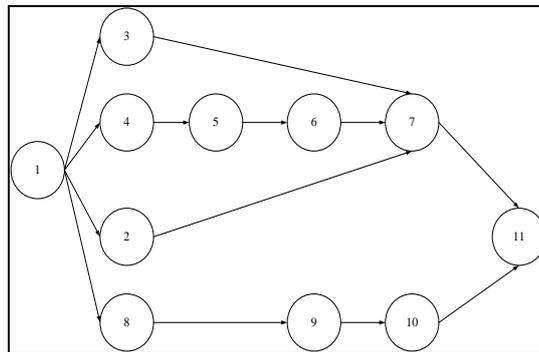
Performansi usulan akan diujikan dengan menggunakan tiga skenario. Skenario 1 yang digunakan untuk menguji cara kerja dari algoritma usulan adalah data Bedworth dan Bailey (1982). Sedangkan skenario 2 dan 3 yang digunakan untuk menguji keandalan dari algoritma usulan adalah set data Thomopoulos (1970) dan Su dan Lu (2007).

Set data yang digunakan pada tahap pengujian algoritma yaitu:

- 1 Set data Bedworth dan Bailey (1982)
Set data Bedworth dan Bailey (1982) dengan dua variasi produk, jumlah elemen kerja (n) = 11, dan waktu siklus (CT) = 10. Set data berupa elemen kerja (j) dan waktu elemen kerja bervariasi (t_j) dapat dilihat pada Tabel 1 dan *precedence diagram* pada Gambar 3.
- 2 Set data Thomopoulos (1970)
Pada set data Thomopoulos (1970) dengan 3 variasi produk, jumlah elemen kerja (n) = 19, dan jumlah demand setiap variasi adalah $d1 = 120$, $d2 = 60$, dan $d3 = 40$. Data elemen kerja (j) dan waktu elemen kerja (t_j) serta *precedence diagram* dapat dilihat pada Tabel 2 dan Gambar 4.
- 3 Set data Su dan Lu (2007)
Set data Su dan Lu (2007) dengan 3 variasi produk, jumlah elemen kerja (n) = 17, dan jumlah demand setiap variasi adalah $d1 = 40$, $d2 = 10$, dan $d3 = 30$. Data elemen kerja (j) dan waktu elemen kerja (t_j) dapat dilihat pada Tabel 3 dan *precedence diagram* pada Gambar 5.

Tabel 1. Set Data Bedworth dan Bailey (1982)

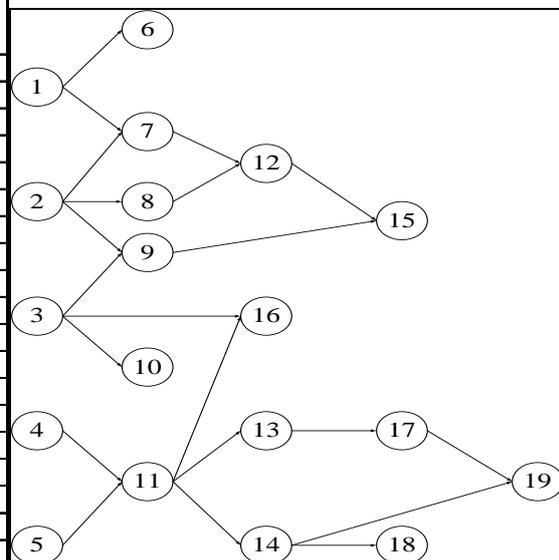
j	t_z		rata-rata t_j
	t_1	t_2	
1	1	1	1.0
2	5	0	2.5
3	4	4	4.0
4	0	1	0.5
5	0	5	2.5
6	0	6	3.0
7	2	2	2.0
8	4	0	2.0
9	3	3	3.0
10	0	5	2.5
11	3	3	3.0



Gambar 3. Precedence Diagram Gabungan Bedworth dan Bailey (1982)

Tabel 2. Set Data Thomopoulos (1970)

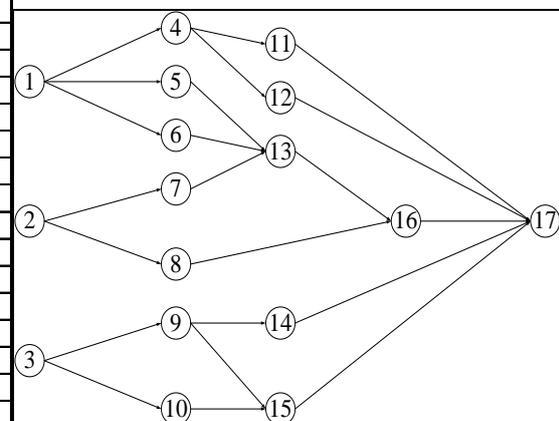
j	t_z			rata-rata t_j
	t_1	t_2	t_3	
1	0.50	0.00	1.00	0.50
2	0.40	0.80	1.20	0.80
3	0.00	0.20	0.40	0.20
4	0.40	0.00	0.00	0.13
5	0.20	0.20	0.20	0.20
6	0.20	0.00	0.00	0.07
7	0.40	0.50	0.60	0.50
8	0.00	0.50	0.50	0.33
9	0.40	0.30	0.20	0.30
10	0.00	0.00	0.20	0.07
11	0.30	0.30	0.30	0.30
12	0.10	0.30	0.50	0.30
13	0.10	0.00	0.10	0.07
14	0.20	0.20	0.20	0.20
15	0.70	1.00	1.50	1.07
16	0.00	0.10	0.00	0.03
17	0.50	0.50	0.00	0.33
18	0.30	0.50	0.30	0.37
19	0.40	0.30	0.00	0.23



Gambar 4. Precedence Diagram Gabungan Thomopoulos (1970)

Tabel 3. Set Data Su dan Lu (2007)

j	t_z			rata-rata t_j
	t_1	t_2	t_3	
1	2.00	2.00	3.00	2.33
2	0.00	3.00	0.00	1.00
3	0.00	0.00	2.00	0.67
4	2.00	0.00	0.00	0.67
5	1.00	2.00	1.00	1.33
6	1.00	2.00	1.00	1.33
7	0.00	2.00	0.00	0.67
8	0.00	5.00	0.00	1.67
9	0.00	0.00	1.00	0.33
10	0.00	0.00	3.00	1.00
11	5.00	0.00	0.00	1.67
12	2.00	0.00	0.00	0.67
13	3.00	1.00	1.00	1.67
14	0.00	0.00	2.00	0.67
15	0.00	0.00	4.00	1.33
16	3.00	1.00	1.00	1.67
17	2.00	1.00	1.00	1.33



Gambar 5. Precedence Diagram Gabungan Su dan Lu (2007)

4.2 Hasil Cara Kerja Algoritma

Perbandingan hasil cara kerja algoritma dapat dilihat pada Tabel 4 berikut ini.

Tabel 4. Perbandingan Hasil Algoritma Usulandengan Penelitian-Penelitian yang Dipublikasikan

Set Data	Penelitian	Jumlah Elemen Kerja (n)	Waktu Siklus (CT)	Jumlah Stasiun Kerja (i)	Efisiensi Lintasan (EL)	Smoothness Index (SI)
Bedworth & Bailey (1982)	Haq et. al (2005)	11	10.00	3	86.67%	2.06
	Rahayu (2011)	11	10.00	3	86.67%	2.06
	Model Usulan	11	9.50	3	91.23%	2.06
Thomopoulos (1970)	Thomopoulos (1970)	19	2.20	3	90.91%	0.07
	Rahayu (2011)	19	2.20	3	90.91%	0.07
	Model Usulan	19	2.00	3	100%	0.00
Su dan Lu (2007)	Su dan Lu (2007)	17	6.00	4	83.33%	
	Rahayu (2011)	17	6.00	4	83.33%	1.89
	Model Usulan	17	5.00	4	100%	0.00

Tabel 5. Perbandingan Hasil Algoritma Usulan Untuk Setiap Pembatas Threshold

Set Data	Batas Threshold	Jumlah Elemen Kerja (n)	Waktu Siklus (CT)	Jumlah Stasiun Kerja (i)	Efisiensi Lintasan (EL)	Smoothness Index (SI)
Bedworth & Bailey (1982)	5%	11	7.00	4	92.86%	1.58
	15%	11	9.50	3	91.23%	2.06
	25%	11	9.50	3	91.23%	2.06
Thomopoulos (1970)	5%	19	2.00	3	100%	0.00
	15%	19	2.00	3	100%	0.00
	25%	19	2.00	3	100%	0.00
Su dan Lu (2007)	5%	17	5.00	4	100%	0.00
	15%	17	5.00	4	100%	0.00
	25%	17	5.00	4	100%	0.00

4.3 Analisis

Berdasarkan Tabel 4 Algoritma *Guided GRASP With Fixed Threshold* memiliki keandalan dalam merancang lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja dengan performansi maksimisasi efisiensi lintasan. Pada penelitian Haq et. al. (2006), set data Bedworth dan Bailey (1982) diselesaikan menggunakan metode *hybrid genetic approach (Hybrid GA)*. Algoritma usulan *Guided GRASP With Fixed Threshold* dapat menghasilkan jumlah stasiun kerja yang sama sebanyak 3 stasiun kerja untuk set data Bedworth dan Bailey (1982) dengan jumlah elemen kerja (n) = 11, CT = 9.5, dan EL yang lebih besar dengan nilai 91,23%.

Pada skenario 2, algoritma usulan *Guided GRASP With Fixed Threshold* dapat menghasilkan jumlah stasiun kerja yang sama sebanyak 3 stasiun kerja untuk set data Thomopoulos (1970) dengan jumlah elemen kerja (n) = 19 dan CT = 2,00. Waktu siklus yang menjadi lebih kecil menghasilkan nilai efisiensi yang lebih besar dengan nilai 100%. Penelitian Thomopoulos (1970) menggunakan set data Thomopoulos (1970) diselesaikan dengan metode *mixed model line balancing with smoothed station assignments*.

Pada skenario 3, penelitian Su dan Lu (2007) diselesaikan dengan metode *genetic algorithm* (GA), sedangkan dengan menggunakan algoritma usulan *Guided GRASP With Fixed Threshold* dapat menghasilkan jumlah stasiun kerja (i) = 4 dengan jumlah elemen kerja (n) = 17 dan $CT = 5$. Waktu siklus yang awalnya 6 menjadi 5 menghasilkan nilai efisiensi lintasan yang semakin besar.

Penelitian Rahayu (2011) diselesaikan dengan menggunakan metode *Guided GRASP* dengan menguji 3 set data yaitu set data Bedworth dan Bailey (1982), Thomopoulos (1970), dan Su dan Lu (2007). Masing-masing menghasilkan jumlah stasiun kerja secara berturut-turut yaitu 3, 3, dan 4. Penelitian dengan menggunakan metode dari pengembangan model *Guided GRASP* yaitu *Guided GRASP With Fixed Threshold* menghasilkan jumlah stasiun kerja yang sama dengan hasil penelitian yang dilakukan oleh Rahayu (2011) bahkan algoritma *Guided GRASP With Fixed Threshold* menghasilkan nilai efisiensi lintasan (EL) yang lebih besar dan *smoothness index* (SI) yang lebih kecil dari penelitian sebelumnya.

Pada Tabel 5 dapat dilihat rekapitulasi hasil perhitungan 3 set data menggunakan metode *Guided GRASP With Fixed Threshold* dengan % *threshold* yang berbeda yaitu 5%, 15%, dan 25%. Dari ke tiga set data tersebut terdapat perbedaan pada hasil jumlah stasiun kerja (i), nilai efisiensi lintasan (EL), dan *smoothness index* (SI) pada set data Bedworth dan Bailey (1982). Hal ini disebabkan karena rentang yang dihasilkan oleh nilai *threshold* 5% kecil sehingga solusi-solusi yang bisa masuk untuk menjadi solusi yang lebih baik semakin dibatasi. Jadi, Semakin besar nilai *fixedTh* semakin besar nilai ambang batas yang diijinkan untuk menerima solusi baru sebaliknya Semakin kecil nilai *fixedTh* semakin kecil penerimaan solusi baru.

Dilihat dari penelitian sebelumnya yang dilakukan oleh Rahayu (2011), terdapat perbedaan pada proses *local search*. Pada penelitian ini, *local search* dilakukan secara bertahap dengan iterasi yang telah ditentukan dan menggunakan nilai *threshold* 5%, 15%, dan 25%. Pada tiap iterasi proses *exchange* dan *insert* dilakukan secara bertahap dimulai dengan melakukan proses *exchange*. Kemudian setelah melakukan proses *exchange* lakukan perhitungan untuk mencari besarnya efisiensi lintasan (EL) dan *smoothness index* (SI) dari lintasan yang telah mengalami perubahan. Nilai efisiensi yang diperoleh dibandingkan dengan batas nilai *threshold*. Selanjutnya lakukan proses *insert* dengan tahap-tahap yang sama dengan proses *exchange*. Kemudian lintasan akhir dari iterasi tersebut dibandingkan dengan lintasan awal atau sebelumnya, apakah lintasan hasil iterasi tersebut lebih baik dibandingkan dengan jalur lintasan sebelumnya dilihat dari besarnya efisiensinya. Apabila mengalami kenaikan maka jalur lintasan yang dipakai adalah jalur lintasan akhir yang didapat dari proses iterasi tersebut ataupun sebaliknya. Lanjutkan langkah-langkah tersebut sampai iterasi terakhir dan didapatkanlah jalur lintasan yang terbaik dengan nilai EL dan SI yang kecil.

4. KESIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari penelitian pengembangan algoritma *Guided GRASP With Fixed Threshold* adalah sebagai berikut:

1. Perancangan lintasan yang dihasilkan adalah model lintasan perakitan *mixed model* dengan kriteria minimisasi jumlah stasiun kerja dengan performansi maksimisasi efisiensi lintasan. Model ini dikembangkan menggunakan algoritma *Guided Greedy Randomized Adaptive Search Procedures (Guided GRASP) With Fixed Threshold*.
2. Pengujian algoritma pada Skenario 1 menggunakan set data Bedworth dan Bailey (1982). Pengujian ini mendapatkan jumlah stasiun kerja (i) yang sama dengan

- penelitian Haq et al. (2006) yaitu berjumlah 3 dengan waktu siklus (CT) yang berbeda sebesar 9.5 menit sehingga menghasilkan nilai efisiensi yang besar yaitu 91,23%.
3. Pengujian algoritma pada Skenario 2 menunjukkan bahwa algoritma usulan dapat menghasilkan jumlah stasiun kerja (i) yang sama dengan set data Thomopoulos (1970) yaitu berjumlah 3 dengan waktu siklus (CT) sebesar 2 menit. Waktu siklus yang menjadi lebih kecil menghasilkan efisiensi yang lebih besar yaitu 100%.
 4. Pengujian algoritma pada Skenario 3 menunjukkan bahwa algoritma usulan dapat menghasilkan jumlah stasiun kerja (i) yang sama dengan set data Su dan Lu (2007) yaitu berjumlah 4 dengan waktu siklus (CT) sebesar 5 menit. Waktu siklus yang menjadi lebih kecil menghasilkan efisiensi yang lebih besar yaitu 100%.
 5. Algoritma *Guided GRASP With Fixed Threshold* menghasilkan solusi yang lebih baik dari penelitian sebelumnya yang dilakukan oleh Rahayu (2011) yang menghasilkan nilai EL yang lebih besar untuk 3 data yang diujikan yaitu berturut-turut 90,23%, 100%, dan 100% dan SI yang lebih kecil secara berturut-turut yaitu 2,06, 0, dan 0.
 6. Semakin besar nilai $fixedTh$ semakin besar nilai ambang batas yang diijinkan untuk menerima solusi baru sebaliknya Semakin kecil nilai $fixedTh$ semakin kecil penerimaan solusi baru.

Saran untuk pengembangan penelitian selanjutnya adalah

1. melakukan perancangan lintasan perakitan *mixed model* dengan menggunakan metode lain yang belum pernah digunakan seperti algoritma *tabu search* dan *ant colony systems*.
2. Pengembangan penelitian selanjutnya dapat juga dilakukan dengan membuka salah satu asumsi, sehingga sistem yang dibahas akan berbeda dengan penelitian ini.

DAFTAR PUSTAKA

- Becker, C., dan Scholl, A., 2006, *A Survey on Problems and Methods in Generalized Assembly Line Balancing*. *European Journal of Operational Research*.
- Bedworth, D. D. and Bailey, J.E., *Integrated Production Control Systems*. Wiley, New Work, 1982.
- Erel, E., dan Sarin, S. C., 1998, *A Survey of The Assembly Line Balancing Procedures*. *Production Planning and Control*.
- Feo, T. A., Resende, M. G. C., 1995, *Greedy Randomized Adaptive Search Procedures*. *Journal of Global Optimization*.
- Gokcen, H., dan Erel, E., 1998, *Binary Integer Formulation for Mixed-model Assembly Line Balancing Problem*. *Computers ind. Engng*.
- Ponnambalam, S. G., Aravindan, P., dan Naidu, G. M., 1999, *A Multi-objective Genetic Algorithm for Solving Assembly Line Balancing Problem*. *International Journal of Advance Manufacturing Technology*.
- Scholl, A., 1999, *Balancing and Sequencing of Assembly Lines*. Second Edition. Physica-Verlag Heidelberg New York.
- Scholl, A., dan Voß, S., 1996, *Simple Assembly Line Balancing – Heuristic Approaches*. *Journal of Heuristics*.
- Su, P., dan Lu, Y., 2007, *Combining Genetic Algorithm and Simulation for the Mixedmodel Assembly Line Balancing Problem*.
- Thomopoulos, N. T., 1970, *Line Balancing-Sequencing For Mixed-Model Assembly*. *Management Science*.
- Thomopoulos, N. T., 1970, *Mixed Model Line Balancing With Smoothed Station Assignments*. *Management Science*.